*** Introduction

Welcome to "Atari World"! This program allows you to create three - dimensional color images, project them in true perspective on the Atari screen, rotate them, move closer, farther away, and many other things.

You may draw, for example, the house of your dreams. Then you can view it as it would be seen from 10,000 feet, or you can fill the screen with a doorknob. You may move from room to room by commanding position changes at the Atari keyboard.

You may make the house spin around repeatedly with a few keystrokes. If you decide to change the size or color of a room, you simply switch to the built - in text editor and enter the changes. The image is promptly changed to suit your wish.

Does this sound like science fiction? You won't think so when you have drawn your first image and have seen it rotate vertically or horizontally, allowing you to inspect it from all possible angles.

"Atari Space" is a cube of mathematical space, 65,536 units on a side, created by this program. You may fill it with anything your imagination and creativity will permit. You may use any of the Atari high - resolution colors. Because the program is written in fast assembly language, the graphic images are created very quickly. In many cases animation effects can be achieved.

You will be shown how to enter information about images, then you will see them appear on the screen. You may then save the image text file to disk for future use or modification. You may also save any specific screen images created, to use as part of other programs or to print on a peripheral printer.

Special parts of images (circles or special mathematical shapes) may be created using BASIC programs, then inserted in a developing image file. Individual files may be combined into a master image.

A library of frequently - used images (example: chairs) may be made, then inserted when needed during creation of an image.

In the following instructions, the sign "***" is used to point out topics of special interest. This makes it possible to scan the text quickly, looking for specific items. For example:

*** Use of the text editor

All images are created using the built - in text editor. The text editor is quite easy to use (much easier than the one available for BASIC programming). So let's get started!

*** Starting Up

To begin, place the program diskette in your disk drive, turn it on, and turn on the Atari. The following menu will appear:

```
***************************************
***          ATARI WORLD            ***
***************************************


     Atari Translation by C. Kingston
      From APPLE WORLD by P. Lutus

         (c) COPYRIGHT 1981 BY
       UNITED SOFTWARE OF AMERICA
             750 THIRD AVE
            N.Y., N.Y. 10017



1) SELF RUNNING DEMO

2) PROMPTING DEMO

3) THE PROGRAM


HIT A KEY (1-3) :
```

Hit "3" and Return. The following sub-menu will be displayed:

```
**************************************
*                                    *
*           ATARI WORLD              *
*                                    *
*   Translation by C. Kingston       *
*                                    *
* From APPLE WORLD (by P. Lutus)     *
*                                    *
**************************************


OPTIONS -

( E ) EDIT IMAGE FILE
( V ) VIEW IMAGE
( S ) SAVE IMAGE


CHOOSE (LETTER) :
```

In normal program use, one uses the (E) Edit option to create an image, then the (V) View option is used. For this example, press "E". The editor prompt lines will then appear at the top of the screen:

*** Editor Entry Format

```
ESC)MOVE  CURSOR.  V)IEW S)TRING B)EG E)ND I)NSRT
T)AKE K)EEP DEL L)INE W)ORD Q)UIT
```

These lines will always appear at the top of the screen while the editor is in use. They serve to remind the user of the available editor options. The letters to the left of the parentheses (example: "V)IEW") are ConTRoL characters.

*** ConTRoL Characters

ConTRoL characters are gotten by pressing the ConTRoL key (at keyboard left), then, while holding it down, pressing the specified letter key.

The bottom line of the editor prompt is a tabulation index for the image entry format, to be described later.

*** Erasing Memory

There may be some nonsense characters on the screen when the editor is first entered. To clear the screen (and the text file memory), press ConTRoL - D. This means to press the ConTRoL key, the SHIFT key, and the "P" key. This function requires the pressing of three keys because it erases memory, something you won't want to happen often. A prompt will appear:

```
ERASE MEMORY (Y/N) ?
```

This is further protection against accidental memory erasure. Having responded with "Y", you will be presented with a clear screen (and file). Now try typing something. The first thing you will notice is that, when you press the space bar, the cursor (flashing square) moves over farther than one space. This is part of the special image input format. For the following examples, place a SEMICOLON (;) at the beginning of each line you type. This will permit normal typing. Now type the following example:

```
; NOW IS THE TIME
; FOR ALL GOOD MEN TO COME
; TO THE AID OF THEIR COUNTRY.
```

If you make a mistake, you may press the left arrow key to delete it. But suppose you wanted to change the word "MEN" to "PEOPLE" in order not to win the enmity of Gloria Steinem. You are already past that word. How shall you change only the word "MEN", without disturbing the other words?

*** Cursor Control

Here's how: Press the ESCape key (keyboard upper left). The cursor now has a "+" on it, and the top line prompt has changed to:

```
CURSOR CONTROL ^ V < >
```

This means that, by using the letter keys I,J,K,M you may move the cursor anywhere on the screen without disturbing the text. Once you have moved the cursor where you want to insert or delete characters, just begin typing again. Now we change the word "MEN" to "PERSONS" (the cursor is marked [ ]):

*** Replacing a Word

(1) Press ESCape (keyboard upper left)

(2) Move the cursor to the desired line, using "^":

```
; NOW IS THE TIME
; FOR ALL GOOD MEN TO COME[+] +--- cursor
; TO THE AID OF THEIR COUNTRY.
```

(3) Move the cursor to the word "MEN" using "<":

```
; NOW IS THE TIME
; FOR ALL GOOD MEN[+] TO COME
; TO THE AID OF THEIR COUNTRY.
```

(4) Delete the word "MEN" with the Backspace key:

```
; NOW IS THE TIME
; FOR ALL GOOD [ ] TO COME
; TO THE AID OF THEIR COUNTRY.
```

(5) Type "PERSONS":

```
; NOW IS THE TIME
; FOR ALL GOOD PERSONS[ ] TO COME
; TO THE AID OF THEIR COUNTRY
```

*** Other Deletion Options

You may also delete characters a word at a time by pressing ConTRoL W, or a line at a time by pressing ConTRoL L. Try it!

*** Character Recovery

All characters deleted using these features may be gotten back by pressing the ConTRoL - F key. This means that you may change the order of words or lines by deleting at the original position, moving the cursor, then recovering the characters using the ConTRoL - F key. Try this also.

If you delete more than 256 characters without recovering any, the oldest deletions will begin to be lost. The most recent 256 deleted characters are saved for you.

*** Page Display Control

If you type a great deal, the text will scroll upward, so that some text will be above the top of the screen. To get back to that text, simply move the cursor to the top of the screen. The text will scroll downward to keep the cursor on the screen, all the way to the beginning of the file if necessary.

*** Fast Scrolling

To move up 12 lines at a time, press ConTRoL R. To move down 12 lines at a time, press ConTRoL C. To jump to the file beginning, press ConTRoL B. To jump to the file end, press ConTRoL E.

*** ConTRoL S (String)

Let us say that you have a long file, and want to find something without scrolling completely through the file. To do this, press ConTRoL S (S = String [a group of characters]). The following prompt will appear:

```
ENTER /NOW/TO BE/ :
```

If you just want to find something, and don't need to replace it, simply enter the thing to be sought: /FIND THIS/. If you want to automatically replace something with something else, enter both: /FIND THIS/REPLACE WITH THIS/. The "/" separates the search string from the replacement. Any character may be used instead of "/", but the character used may not be part of either entry.

After you have made your entry, the program will search FORWARD through the file for the search string. This means that you must be at the file beginning to catch every occurrence of the string. Another prompt appears while the search is underway:

```
CARR RET=MORE CTRLR=REPLACE TYPE=QUIT
```

This means to press RETURN to continue the search, press ConTRoL R to replace the displayed string with the specified replacement, and

simply type normally to exit the string function. Example:

You have typed:

```
; SACRAMENTO IS THE CAPITAL OF CALIFORNIA
```

After another 10,000 lines, you remember that you have misspelled "CAPITOL". To find it, you:

(1) Jump to the file beginning, using ConTRoL B.
(2) Press ConTroL S, and enter: /CAPITAL/CAPITOL/
(3) The program will find the error and stop. It is waiting for you to decide to replace, go on, or quit. If you press ConTRoL R, the replacement will be made, and the search will continue. At the end of the search, the cursor will be positioned at the end of the file.

*** Image Entry

The preceding examples have used words for clarity. In practice, the editing functions are used most often to edit lists of numbers, the main entry mode for the text editor.

If an entry is made without a semicolon at the line beginning, the program will try to interpret it as an image segment. Try entering some numbers without a semicolon at the line beginning. If you press the space bar, you will notice that the cursor will jump to align itself with the categories listed in the editor bottom line. These categories will now be explained.

*** Cartesian Coordinates

The program accepts image information in three dimensions using an entry format called "Cartesian Coordinates". This simple system consists of three numbers: X (left/right), Y (up/down), and Z (forward/backward). Cartesian

Coordinates allow you to define a specific position in three - dimensional space.

Suppose a friend wanted to tell you where he lived in a big city. He might say "Start at the bus station. Walk East 10 blocks, then walk North 12 blocks, then take the elevator to the 7th floor."

If your friend knew Cartesian Coordinates, he could have said "The bus station is located at X = 0, Y = 0, Z = 0. Move along X until it is equal to 10. Move along Z until it is equal to 12. Move along Y until it is equal to 7. Knock on my door."

You make entries to the program in just this way. Each time you enter a new line in the text editor, the program will draw a new line on the screen (unless you tell it not to). Don't forget to hit return (CR) at the end of the line. Here's how the entries are made:

*** Sample Entry

```
COMMAND COLOR    X         Y         Z
NA      3        -50       +50       0 (CR)
```

*** Drawing Commands

The extreme left column accepts some letter commands that tell the program what to do with the data in the entry. These letters are:

Command: "N" = New line.

If "N" is present, the program will start a new entry.

If "N" is absent, the program will draw a continuing line from the last entered position to the position specified in this entry.

Command: "A" = Absolute position.

If "A" is present, the program will set the drawing position to the absolute value of the numbers in the entry.

If "A" is absent, the program will ADD the numbers in the entry to the present program values, and then draw a line to the result.

This entry:

```
A                -100      -100      -100
```

means "Draw a line from the previous position to the absolute position X = -100, Y = -100, Z = -100".

This entry:

```
[no letter]      -100     -100      -100
```

means "Add (-100) to the present X, Y, and Z values, and draw a line from the previous position to the result".

Command: "Q" = Quit drawing. Skip the rest of the file (including the present entry). This function is used to selectively draw parts of an image.

All of these letters, or none, may be present in an entered line.

*** Drawing an Image

Now we will draw something. First, clear the screen using ConTRoL - D, as in the beginning example. Then enter the following (the comments are not required, but may be entered into the file as reminders):

```
COMMAND COLOR    X        Y        Z
; SET THE (LOWER LEFT) BEGINNING OF A SQUARE:
;
NA      3        -50      -50      -50
;
; MEANS " START A NEW LINE,
; BEGINNING AT THE ABSOLUTE POSITION
; X= -50, Y= -50, Z= -50, WITH A
; COLOR OF 3 (WHITE)"
;
; NOW DRAW THE SQUARE:
;
; INCREASE X BY 100 (MOVE RIGHT)
;
                 +100
```

(Note: If there is no entry in a column (such as X, Y, or Z), the previous value is used. Therefore, to keep an old value, press the space bar without making any entry in that column.)

```
COMMAND COLOR    X        Y        Z
;
; INCREASE Y BY 100 (MOVE UP)
;
                          +100
;
; DECREASE X BY 100 (MOVE LEFT)
;
                 -100
;
; DECREASE Y BY 100 (MOVE DOWN)
;
```

```
                          -100
```

*** Viewing an Image

These entries will draw a square on the screen. To see the square, we will use ConTRoL V (View). Press ConTRoL V now. If this is the first use of the viewing feature, you will be asked to enter:

*** Setting the view position

```
( A ) ABSOLUTE SETTINGS:

X POSITION      :
Y POSITION      :
Z POSITION      :

SCALE           :

HORIZONTAL ANGLE:
VERTICAL   ANGLE:

(64 = 90 DEGREES)

(NO ENTRY) = USE DISPLAYED VALUE
( S ) = START AGAIN, ( Q ) = QUIT.
```

There will be some numbers on display for each of these entries. They will probably be random numbers, if this is the first use of the viewing feature. Enter the following values:

```
( A ) ABSOLUTE SETTINGS:

X POSITION       : 0
Y POSITION       : 0
Z POSITION       : 0

SCALE            : 400

HORIZONTAL ANGLE: 0
VERTICAL   ANGLE: 0

(64 = 90 DEGREES)

(NO ENTRY) = USE DISPLAYED VALUE
( S ) = START AGAIN, ( Q ) = QUIT.
```

The reason the previous values are displayed is to let you know what value is presently in use. If the value is satisfactory, you may simply press RETURN without making any entry. This will preserve the displayed value.

While entering changes, you also may capture part of the previous value by pressing the right arrow key so that the cursor passes across the digits.

After the last number has been entered, the screen will show a white square. This is the image you entered.

*** Rotating the image

*** Numeric view entries

While viewing the image, press "1" on the keyboard. When you do this, the following prompt will appear at screen bottom:

```
NUMBER:1
```

The "1" displayed is the digit you pressed. Now press "0". "10" should be on display.

Now press "H" (horizontal angle of rotation). The square on the screen should rotate a little. Pressing "H" repeatedly will make the square rotate completely around. To rotate the square vertically instead of horizontally, enter a number (such as was done above) followed by the letter "V" (vertical angle of rotation). The square will now rotate vertically at each press of "V".

*** View command memory

There is no need to re-enter a number for each press of "H" or "V". They will remember the last number that was entered for them. To change their number (angle of rotation), simply enter a new number and press "H" or "V" again. They will be re-programmed with the new number.

To rotate in the opposite direction, enter a number preceded by "-".

*** View Control Commands

This is a list of the characters that control the view options (also see Appendix B: Short - form listing of view commands). Remember that, unlike the editor, these are ordinary characters, not ConTRoL characters:

[number] H: Change the horizontal view angle by the value of [number] (64 = 90 degrees).

Example: The entry "10H" will change the horizontal angle by 14 degrees. Negative numbers are allowed for all entries.

[number] V: Same as above for the vertical view angle.

[number] X: Move the view position along the X axis by the value of [number].

[number] Y or Z: Same as above in the Y and Z axes.

[number] S: Adjust the viewing scale by the value of [number]. The use of scale is explained later.

P: Display the present viewing position, scale, and angles, without accepting entries. This display is more convenient to use than the following (A) if no changes are contemplated.

A: Display as above, but accept changes.

[number] R: Repeat the command that follows [number] times.

An example of the "R" Repeat function

Enter: "10H". This programs the horizontal angle to move in steps of 14 degrees (an entry of 64 is equal to 90 degrees).

Enter: "10RH". This will repeat the "H" command 10 times and then stop. If you press any key during the 10 repeats, the movement will stop.

D: Watch the image being drawn. Pressing this key switches the displayed screens so that the viewed screen is also the screen being drawn. Pressing "D" once again returns to normal operation.

E: Return to the editor.

If you get lost while moving the view position around, simply press "A" (Absolute position) and re-enter the beginning positions listed above. The "A" Absolute position entry overrides the changes created by the keyboard entries described above.

After you have rotated the square to your satisfaction, press "E" to return to the editor. The editor display will return, with the cursor in the original position.

You may have noticed that the square wasn't rotating on center. It was offset from the center of the screen. This is because there was an offset intentionally entered in the third (Z) dimension as the shape was begun. This is because we are going to extend our square into a cube.

Jump to the file end using ConTRoL E (if you aren't there already), and enter:

```
COMMAND COLOR    X        Y        Z
; INCREASE Z BY 100
;
                                   +100
;
; NOW DRAW ANOTHER SIDE OF THE CUBE
;
; INCREASE X BY 100
;
                 +100
;
; INCREASE Y BY 100
;
                          +100
;
; DECREASE X BY 100
;
                 -100
```

```
;
; DECREASE Y BY 100
;
                          -100
```

View the image now by pressing ConTRoL V. If necessary, you may reset the absolute positions listed earlier by pressing "A" and making the entries needed.

It can be seen that we don't have a complete cube yet. Three lines are still missing. Now we complete the cube:

```
COMMAND COLOR    X        Y        Z
;
; DRAW 3 NEW LINES
;
N                +100
                                  -100
N                         +100
                                  +100
N                -100
                                  -100
```

Do you see why these last entries complete the cube? Try thinking in three dimensions to visualize it, or view individual lines as they are entered to see how the shape is built up. This visualization process will help you in the use of this program.

*** View Colors

Noow that you know how to enter shapes, experiment a little. Try changing the color of the cube, using the following table:

Color No.	Color
0	Black (same as background)
1	Orange
2	Blue
3	White

The Color No. is entered in the editor mode in column 2.

*** Editing while viewing

Place the cursor next to the color entry at the file beginning, and change the color. Then observe the results using the "V)IEW" function. Each time you return to the editor, the cursor will be in the position at which you left it.

*** Automated List Processing

Here is an example of the use of the string replacement function in a number list. Let us say that we want to increase the size of the cube, and don't want to replace each number individually. Do the following.

(1) Jump to the file beginning, using Ctrl-B.
(2) Press Ctrl-S (S = String), and enter:

```
/100/200/
```

(3) After this entry, the program will find the first "100" in the file and wait for your response. Now replace all of the "100"s with "200" by pressing Ctrl-R repearedly. To save time, you may hold it down to activate the auto repeat.

(4) Repeat the entire proceedure above, this time using the string entry "/50/100/".

Now the cube will be twice as large. It may even exceed the screen boundaries in some positions.

*** Saving the text file

Suppose you want to save the text file to disk. To do so, press Ctrl-K (K = Keep). You will be transfered to the mini-DOS with the following menu:

```
    (S) SAVE          (L) LOAD
    (D) DELETE        (R) RENAME
    (A) DIRECTORY     (RET) RETURN

    ENTER (LETTER) :
```

Put your own disk in the drive you are going to use to save the file. DO NOT REMOVE THE WRITE PROTECT TAB FROM, OR SAVE ANY FILE TO, THE MASTER DISK! To save a file naed "TEST" to drive 2, press "S". You will get the prompt:

```
    ENTER FILE NAME :
```

Type "D2:TEST.FIL" and press RET. You do not need to use the drive number if you are only using drive 1. You can use any extension that you wish.

*** Disk commands

The mini-DOS allows you to Load a file, Rename a file, and Delete a file. You may also list the directory for drive 1 by pressing "A". Return to the editor by pressing RET alone.

*** File Insertion

Let us suppose that you had drawn a chair and want to insert 30 copies of it into a developing image. Rather than typing in the coordinates over and over, simply draw the chair (using only relative positions - no "A" commands) and save it to disk with an appropriate title (we'll use "CHAIR" here).

Later, while drawing the main image, you may simply:

(1) Position the cursor in the file where you want the chair to go.
(2) Press Ctrl-I
(3) You will be prompted with
"ENTER FILE NAME :"; enter "CHAIR"

Return to the editor with RET. You will see that the chair file will have been inserted at the cursor position. This may be done repeatedly.

*** File Markers

If you need to save part of a file, put your cursor at the end of the section you wish to save; type Ctrl-T (Take). The Atari will prompt with "FILE MARKER :". Type a character, or set of characters that indicates the start of the section that you wish to save. The Atari will then prompt "ENTER FILE NAME :". Type in what you want to name the file and it will be saved under that name.

*** Relative Positioning

It is for this insertion capability that the RELATIVE POSITION format is available. At each insertion of a chair, a chair will be drawn AT THAT RELATIVE POSITION. Fof example, to draw a row of chairs, you would:

(1) Move to the X, Y, Z position for the first chair.
(2) Insert the "CHAIR" file.
(3) Move to the next chair position.
(4) Repeat steps (2) and (3) as required.

There is a file on the Master Disk named "OFFICE" that shows the result of this proceedure.

*** Saving a screen image

After viewing a screen image (not a text file), you may want to save it for use in annother program, or to print it on a printer. To do so, press "Q" to exit the View mode. You will go to the main menu. Now press "S". The image will be displayed, with the prompt "IMAGE FILE NAME :" at the bottom of the screen. Enter the file name and the image will be stored on the disk.

Technical note: The image is always saved from Screen 1 ($6000-$8000) for uniformity. The image is moved there automatically by the program.

*** Advanced Concepts

The information from this point is not essential to the opperation of Atari World, but makes possible the fullest use of the available features.

Some technical terms will clarify the following discussion: Moving along the X,Y, or Z axes is called TRANSLATION. Stepping through horizontal and vertical angles is called ROTATION. The X,Y, and Z view position is called the ORIGIN. The window through which we see the images is called the VIEWPORT.

*** Scale and Translation

(1) By translating along the axis toward which

the viewport is "facing". Example: if the horizontal and vertical angles are both 0, then we are looking along the Z axis. Therefore changing the position along the Z axis will increase or decrease image size, whereas movement along the X and Y axes will simply move the image horizontally and vertically.

(2) By changing scale. A scale entry does not change the X, Y, and Z position (the "origin"), it only changes an imaginary rod that is the length of the scale, and extends from the origin out along the entered horizontal and vertical angles (see VIEWDEMO to visualize this). The viewport is at the end of this imaginary rod, and rotates as angles are entered.

Setting a view size through translation has the disadvantage that, if a rotation is later entered, the image will rotate out of the field of the viewport, because the origin position is now distant from the object being viewed.

On the other hand, using the scale number to change image size makes it possible to rotate the image completely, while keeping it centered in the viewport.

The origin (X, Y, and Z positions) setting is best used to "point" to an object. The object then may be rotated around using angle changes, or made larger or smaller through the use of scale.

*** Number range

The mathematical package included in Vector World will handle numeric entries of -32767 to +32767 along the three axes X, Y and Z. This means that there are about 281 TRILLION definable points in Vector Space. The following image characteristics and limitations are

important to remember:

*** Text file memory size, capacity

(1) The text file memory size is 8,200 characters long. This means that about 600 average line descriptions may be entered. If memory fills during editing, a bell will ring and no more entries will be accepted. The file may be saved even though a memory overflow has been indicated.

*** Image conversion formulas

(2) The algorithms are all written in assembly language for the sake of speed, including those that perform trigonometric calculations. The following formulas are used to convert the image for viewing:

```
PY = PLOT Y. PX = PLOT X. PF = PLOT FACTOR
(DEPTH)
VPOS = VIEW POSITION
[H] = HORIZONTAL ANGLE. [V] = VERTICAL ANGLE
BEGIN:
X = X - VPOSX. Y = Y - VPOSY. Z = Z - VPOSZ
PY = Y*COS[V]+SIN[V]*(X*SIN[H]-Z*COS[H])
PX = X*COS[H]+Z*SIN[H]
PF = Y*SIN[V]+COS[V]*(X*SIN[H]-Z*COS[H])+SCALE
PY = 80 - (PY/PF). PX = 140 + (PX/PF)
END.
```

Because these computations must be performed for each line ending, there is a substantial mathematical overhead not directly involved in line drawing. As a result, a 400 - line image (such as system diskette files "Chalice" and "Globe") takes nearly 4 seconds to update in the worst case (less complex images will update several times a second).

It therefore is a good idea to use line continuations as much as possible, as a new line must have both ends computed, and a continuation needs only one.

*** Numeric limitations

(3) Because fixed point computation is used, it is best to make numeric entries as large as are consistent with drawing scale. Because of truncation during computation, a cube 10 units on a side will be 10% distorted in some positions. A cube that is only 1 unit on a side will flatten out completely in some positions.

Drawing a cube 100 units on a side effectively removes this distortion, since the truncation is then smaller than the display resolution. Smaller objects may be used if they won't be "approached" too closely or too often.

(4) If a cube is drawn 64,000 units on a side, it won't be possible to back up far enough to see it (without "falling into the dragons" at the edge of Vector World). However, it is permitted to draw images of normal proportions that are OFFSET by this distance, and viewing will function normally.

*** To plot or not to plot?

(5) A line that has neither end on the screen will not be drawn. This is to speed up the drawing of large objects in which details will be observed close up. This effect can be seen while "approaching" an image. Lines close to the view position will disappear entirely as the view position is moved forward.

*** Color use

(6) If detail is the most important requirement, USE WHITE. Color lines are, on the average, twice as thick as white lines. This comes about because of the way Atari generates colors. If an

image is to be printed on a peripheral printer, there is no point in keeping the colors, they should be removed to improve detail.

*** Appendix A: Short - form list of editor commands.

Ctrl	E:	Jump to file end
	B:	Jump to file beginning
	R:	Move up 12 lines
	C:	Move down 12 lines
	W:	Delete one word
	L:	Delete one line
Backspace		Delete one character
Ctrl	F:	Recover last deleted character
	S:	String search and replace
	K:	Disk commands
	V:	View image
	D:	Erase text memory
	A:	Add at cursor
ESC		Cursor control (Use arrow keys)

Demonstration Options:

J:	Pause 1 second
G:	Keyboard input
N:	Supresses all text displays in editor and view.
Q:	Quit, return to main menu

*** Appendix B: Short - form list of view functions and controls

[number]: Set value for command to follow
H: Step horizontal angle [number]
V: Step vertical angle [number]
(64 = 90 degrees)

X: Move along X axis [number] units
Y: Move along Y axis [number] units
Z: Move along Z axis [number] units
R: Repeat command that follows [number] times
P: Display program values
A: Set program values
D: Watch drawing. Press "D" again to cancel.
E: Jump to editor
Q: Return to main menu

Number entries are "remembered" by the individual commands that follow, including repeat. You only need to enter a number to CHANGE a previous number for that command.

*** Appendix C: How to create an image file with a BASIC program.

You may use a BASIC program to create an image file; the program "GENERATE" included on your disk will generate solids of rotation when given the profile coordinates.

You should understand the internal format of the image text file before attempting to create one. The following is a breakdown of a file entry:

COMMANDS "N", "A", or "Q" space color space X space Y space Z carriage return

If there is no command entry for Y or Z, the carriage return may be earlier in the line. There must be the correct number of spaces before s given command appears. For example, a line with a single entry for the A axis would be:

space space space space Z

A typical entry line in BASIC might be the following:

```
70 PRINT N$;"A ";C$;" ";X;" ";Y;" ";Z
```

The strings "N$" and "C$" are the new line command and color, both as characters. In this way, they can be set to the null string (N$="") during running of the program. Passing up unused or redundant commands saves time in image processing later.

The following is a listing of a BASIC program that incorporates all of the foregoing. This is the program that was used to create the Chalice and Globe files on the system disk. It is on the master disk under the name "GENERATE.BAS":

```
  5 REM GENERATE
 10 S=20 : REM STEP
 20 DIM H(100) , V(100)
 30 DIM C$(1),N$(1),Y$(10)
 40 OPEN #1,8,0,"D1:DATAFILE"
 50 REM READ PROGILE FROM DATA
 60 FOR E=0 TO 100
 80 READ H,V:H(E)=H:V(E)=V
 90 IF H<>0 OR V<> 0 THEN NEXT E
100 E=E-1
110 C$="3"
230 REM DRAW HORIZ LINES
240 FOR Q=0 TO E
250 N$="N"
260 Y=INT(V(Q)):Y$=STR$(Y)
270 FOR R=0 TO 360 STEP S
280 Z=INT(SIN(R)*H(Q))
290 Z=INT(COS(R)*H(Q))
300 PRINT #1;N$;"A ";C$;" ";X;" ";Y$;" ";Z
310 C$="":Y$="":N$=""
320 NEXT R:NEXT Q
330 REM DRAW CERT LINES
340 FOR R=0 TO 360 -S STEP S
350 N$="N"
360 FOR Q=0 TO E
370 Y=INT(V(Q))
380 X=INT(SIN(R)*H(Q))
390 Z=INT(COS(R)*H(Q))
400 PRINT #1;N$;"A ";C$;" ";X;" ";Y;" ";Z
410 N$=""
420 NEXT Q:NEXT R
```

```
  425 PRINT #1;CHR$(14);CHR$(14);CHR$(14)
  430 CLOSE #1
  440 REM DATA STATEMENTS HERE
  450 REM LAST PAIR MUST BE 0,0
```

This program takes the contor of the side of a shape, entered as data statements, and converts it into a symmetrical 3D image table for Atari World. The image table is written directly to a disk file. Be sure you do not use the master disk for this. Alwayes write files to one of your own formatted diskettes.

Once generated, the files can be Loaded into the Atari World editor, examined, and the viewed with the main program. The following inputs will create a cylinder with cone shapes at each end.

```
460 DATA -1000,1000,-500,500
470 DATA -500,0,-500,-500
480 DATA -1000,-1000,0,0
```

This provides an interesting shape considering the simplicity of the imput data. To increase or decrease the complexity (and drawing time) of a shape created with this program, change the stepping angle constant in line 10.

{end}